Cognitive Estrangement as Method: Mining Invariants with Category Theory & HoTT for Reproducible Insight

Douglas C. Youvan

doug@youvan.com

www.youvan.ai

September 9, 2025

Most AI work narrates results from opaque training; this paper proposes a different posture: use cognitive estrangement as a deliberate method to expose what truly survives representation changes. By “estrangement” we mean a controlled defamiliarization of the workflow: swap training-for-accuracy with mining-for-invariants, replace black-box claims with witnesses and replay, and force each step through structure-preserving transforms E(.) so assumptions become testable objects. Our backbone is Category Theory and HoTT: functors, limits/colimits, adjunctions, paths, and higher equalities supply the grammar for invariants; equivalence, not equality, is the truth predicate. Practically, we give a deterministic, local pipeline that anyone can re-run on commodity hardware: fixed seeds, hashed environments, executable storyboards, and coherence obligations (horn fill, triangle/pentagon checks, pushout sanity) that either pass with proofs or fail with concrete diagnostics. Detectors favor compression and stability under E(.), elevating candidates that “hold” across duals, slices, and retypings. The aim is not to possess truth but to host it: make hypotheses falsifiable, make witnesses public, and make insight reproducible. Readers can treat this as both a manifesto and a kit: a way to do serious inquiry with small machines and big structures.

Keywords: cognitive estrangement, invariants, category theory, homotopy type theory, HoTT, reproducibility, local AI, deterministic seeds, mining vs training, executable storyboards, coherence, horn filling, associativity (triangle/pentagon), pushouts and pullbacks, adjunctions, univalence, interpretability, compression metrics, witnesses and logs, epistemic humility. A collaboration with GPT-5.

1. Thesis & Motivation

1.1 Problem statement

Most AI narratives present “trained” results as if they were knowledge. What they usually lack are: (a) invariants that survive representation changes, (b) public witnesses that let others replay the claim, and (c) a proof-guided discipline that rejects seductive but brittle patterns. Without those, we cannot tell insight from overfit coincidence.

1.2 Working definition: cognitive estrangement (technical)

We use “cognitive estrangement” as a method: deliberately reframe a familiar pipeline so hidden assumptions become inspectable objects. Concretely, we interpose an operator E(.) that applies structure-preserving transforms to data, programs, and intermediate claims, then demand that candidate results remain stable “up to equivalence.” The point is to force defamiliarization just enough to expose what truly holds.

1.3 Proposal in one line

Replace “training for accuracy” with “mining for invariants,” under a reproducible, proof-aware workflow that anyone can re-run locally.

1.4 Backbone concepts (CT / HoTT)

- Category theory gives the grammar for stability: objects, morphisms, functors, limits/colimits, adjunctions, factorization systems, etc.
- HoTT supplies the equality policy: equality is witnessed equivalence; higher paths record how and why two things are the same up to structure.

1.5 Minimal formal scaffold (ASCII-safe)

- E : Rep -> Rep is an endofunctor on a space of representations Rep.
- A “claim” is a pair (r, phi) with representation r in Rep and predicate phi(r)=true produced by a detector.
- An invariant I is a property such that phi(r)=true implies phi(E(r))=true for all admissible E in a chosen family.

- “Witness” means the bundle W = {seeds, logs, transforms, proofs} that allows third parties to replay and test phi and its stability under E(.).

1.6 Contributions (what this paper adds)
C1. A principled use of estrangement as a *method* to surface invariants, not as rhetoric.
C2. A mining pipeline: seeds -> transforms -> constraints -> detectors -> selection, instrumented with witnesses and coherence checks.
C3. A library of E-transforms (duals/opposites, slice/coslice, currying, re-typing, factorization) with pass/fail obligations.
C4. Practical reproducibility: fixed seeds, hashed environments, executable storyboards, and a one-hour replay path.
C5. An ethics posture: claims must be replayable on small machines; no witness, no claim.

1.7 Assumptions and non-goals
A1. We assume access to commodity hardware and a POSIX/Windows toolchain; cloud is optional, not required.
A2. We assume detectors can be biased; stability under E(.) helps filter that bias.
A3. We do not claim “ground truth”; we claim “witnessed stability under declared transforms.”
Non-goals: benchmarking leaderboards, black-box novelty, or dependence on secret data.

1.8 Testable claims (falsifiable)
T1. For a specified E-family, at least one mined result per case study remains stable under all E in that family.
T2. Third parties can reproduce both the positive stability and at least one failure case (where a candidate collapses under E).
T3. Compression-linked detectors (e.g., COMP2) correlate with E-stability better than naive accuracy in our settings.

1.9 Reader’s map
Section 2 clarifies estrangement for research; Section 3 recalls the CT/HoTT primitives we rely on; Section 4 defines E in more detail; Section 5 gives the

mining pipeline; Sections 6–7 cover obligations and detectors; Section 8 provides case studies; Sections 9–14 cover implementation, posture, related work, limits, future work, and the reproducibility kit.

2. Conceptual Frame: Cognitive Estrangement for Science

2.1 Plain-language definition (operational)
Cognitive estrangement, as used here, is a *controlled defamiliarization* of a research workflow so that hidden assumptions become inspectable objects. We do this by interposing transforms E(.) at chosen waypoints and then asking: which claims survive these shifts *up to equivalence*?

2.2 Why estrangement helps

- It breaks tacit couplings (method ↔ metric ↔ narrative) that can smuggle in spurious regularities.
- It sharpens falsifiability: a claim that fails under benign E(.) was likely overfit to representational accidents.
- It produces *witnessable negatives*: when a candidate collapses under E(.), we can show exactly where and why.

2.3 Two audiences, one spine

- Human layer: concise prose, diagrams, and "why this matters now."
- Machine layer: seeds, logs, E-traces, and obligations (horn fill, triangle/pentagon, pushout sanity) that either pass with proofs or fail with diagnostics.
  Estrangement links the two: the same E(.) that makes the prose unfamiliar also drives the machine checks.

2.4 What counts as estrangement (non-exhaustive)

- Category-level: Opp(C), dualization, slice/coslice, change of base, currying/uncurrying.

- Typing/encoding: re-typing objects, refactoring morphisms via factorization systems, alternate normal forms.
- Dataset/program views: permutation-invariant relabeling, schema reindexing, benign augmentations that preserve meaning.
  These are *structure-preserving* in intent; the point is not to randomize, but to reveal invariants.

2.5 Minimal behavioral contract for E(.)
Let Rep be a space of admissible representations and Claim = {(r, phi)} with r in Rep and detector phi.

- Closure: E: Rep -> Rep (stays inside the admissible space).
- Traceability: each application emits an auditable record (E_id, params, hash_in, hash_out).
- Stability test: phi(r)=true implies phi(E(r))=true for all E in a declared family if the property is an invariant.
- Graceful failure: when stability breaks, emit a *counter-witness* (r, E, delta) describing the minimal divergence.

2.6 Estrangement budget
To avoid “over-estrangement,” we parameterize an intensity scale:

- E_light: cosmetic/type-level changes expected to preserve semantics.
- E_medium: categorical duals, slice/coslice flips, factorization rewrites.
- E_strong: compounded transforms that should still preserve invariants but may stress detectors.
  Policy: a candidate must survive all E in {E_light, E_medium} before being promoted; E_strong is used to probe limits.

2.7 How estrangement reframes evaluation
Standard pipelines optimize for task accuracy under one view. Our pipeline asks two questions before accuracy:
Q1: Is there a claim (r, phi) with a public witness?

Q2: Does the claim remain true under declared E(.) families?
Only then do we ask Q3: Among stable candidates, which are simplest/strongest by our detectors?

2.8 Example intuition (no special notation)
Suppose a putative associativity pattern appears in a “triangles-in-squares” storyboard. Apply Opp(C) (reverse arrows) and a slice reindexing. If the pattern reappears—same proof obligations discharging after E(.), same logs differing only by E-traces—we treat the pattern as a mined invariant rather than a coincidence.

2.9 Risks and mitigations

- Risk: estrangement that changes meaning. Mitigation: restrict E to a vetted catalog with proofs-of-preservation where possible.
- Risk: metric brittleness (detectors oscillate under E). Mitigation: prefer compression/stability detectors, report sensitivity bands.
- Risk: cognitive load for readers. Mitigation: pair each E(.) with a one-line “what changed, why it should not matter.”

2.10 Takeaway
Estrangement is not ornament; it is a method. It relocates the burden of proof from single-view performance to multi-view stability with public witnesses.

3. Formal Backbone (CT / HoTT Primitives)

3.1 Categories and functors (minimal kit)

- A category C has Obj(C), Mor(C), id_X, and composition g∘f with associativity and identities.
- A functor F: C -> D maps objects and morphisms, preserving ids and composition.
- Natural transformation eta: F => G is a family {eta_X: F(X) -> G(X)} s.t. G(f)∘eta_X = eta_Y∘F(f).

- Invariants often appear as properties P on objects/morphisms that are preserved by change of view via functors.

3.2 Limits, colimits, and universality

- A cone to a diagram D: J -> C is (N, {phi_j: N -> D(j)}). A limit (lim D) is terminal among cones.
- A cocone is dual; a colimit (colim D) is initial among cocones.
- Pushout is colim of a span A <-s- S -t-> B; pullback is lim of a cospan A -f-> T <-g- B.
- Universality is an invariant: if it truly characterizes a construction, it should remain under admissible E(.).

3.3 Adjunctions and factorization hints

- Adjunction F ⊣ G gives unit eta: Id -> G∘F and counit eps: F∘G -> Id with triangle identities.
- Many “structure-revealing” transforms arise from adjunctions (e.g., currying/uncurrying).
- Orthogonal factorization systems (E,M): every f factors as m∘e with lifting and uniqueness conditions; useful for re-typing/re-factoring without changing essence.

3.4 Opposites, slices, and change of base

- Opposite category C^op reverses arrows; a basic estrangement that should preserve genuine invariants.
- Slice C/X has objects (A -> X) and morphisms commuting over X; coslice X\C dualizes.
- Change of base functors let us view the same pattern relative to a new anchor; invariants should not depend on incidental anchoring.

3.5 HoTT equality policy (why equivalence, not equality)

- In HoTT, equality is an *identity type*: Id_A(x,y) is the type of paths (witnesses) from x to y.
- Propositions are types; proofs are terms. “x = y” means there exists p: Id_A(x,y), not a bare boolean.
- Univalence (informal here): equivalence of types A ≃ B induces equality A = B at the universe level.
- For us: the right truth notion for “same structure” is *equivalence with witness*, not syntactic sameness.

3.6 Coherence and higher structure

- Associativity is governed by pentagon constraints; unit laws by triangle constraints.
- “Horn filling” language: partial boundary data that must be fillable by a cell consistent with stored faces.
- Higher cells record *how* equalities agree; coherence means these higher equalities compose consistently.

3.7 What we call an invariant (paper’s working notion)
Given a family E of admissible transforms on representations Rep and a detector phi: Rep -> {true,false}:

- Stability: phi(r)=true implies phi(E(r))=true for all E in the family.
- Witnessed: there exist concrete paths/proofs (logs, derivations) that certify stability, not just an empirical hunch.
- Functorial resilience: when r is transported along a structure-preserving functor U, the property ascends/descends appropriately (e.g., preserved or reflected by U as declared).

3.8 Executable storyboard vocabulary (bridging CT/HoTT to runs)

- “Square” stands for a commuting diagram; “triangle/pentagon check” stands for basic coherence tests.
- “Pushout sanity”: verify universal property with uniqueness up to chosen equivalence.
- “Counter-witness”: minimal data showing where universality or coherence fails after an E-transform.

3.9 Takeaway
We do not enshrine a single formalism; we carry a small, composable kit: categories/functors, universality, adjunctions/factorizations, opposites/slices, and a HoTT view of equality-as-witness. This is enough to define, test, and *transport* invariants through E(.), which is the core of the method.

4. The Estrangement Operator E

4.1 Informal role
E is the “defamiliarizer.” We insert E at safe points in the workflow to force benign changes of view. A true invariant should survive E; a brittle pattern should not.

4.2 Minimal specification (ASCII-safe)

- Universe of reps: Rep.
- E: Rep -> Rep is an endofunctor on Rep.
- Family: \mathcal{E} = {E_i : Rep -> Rep}.
- Claim: (r, \phi) with r in Rep and detector \phi: Rep -> {true,false}.
- Invariant under \mathcal{E}: \forall E in \mathcal{E}, \phi(r)=true => \phi(E(r))=true.

4.3 Traceability contract
Every application emits a tuple:
(E_id, params, hash_in, hash_out, time, node),

where hashes cover content + environment (seed, versions). Without a trace, an E-step “did not happen.”

4.4 Safety classes (estrangement budget)

- E_light: renamings, re-typings, normal-form rewrites that preserve denotation.
- E_medium: Opp(C) (dual), slice/coslice changes, currying/uncurrying, admissible factorization rewrites.
- E_strong: compositions of the above that may stress detectors while preserving meaning.

Policy gate: promotion requires stability across E_light \cup E_medium; E_strong probes limits and produces counter-witnesses when stability fails.

4.5 Algebraic laws we require (where applicable)

- Identity: there exists Id_E with Id_E(r)=r.
- Composition: E_j \circ E_i is in \mathcal{E}, with recorded order.
- Idempotence (select cases): E \circ E \simeq E (up to declared equivalence).
- Functoriality (when E is categorical): E preserves identities and composition on internal morphisms.

4.6 Preservation vs. reflection
For a functor U: Rep -> View, we document for each E whether U preserves or reflects the property tested by \phi. This prevents false positives caused by a lossy view.

4.7 Catalog sketch (examples we actually use)

- E_op: take opposite category C^op (reverse arrows).
- E_slice[X]: move to slice C/X (or coslice X\C).
- E_curry: curry/uncurry at declared bifunctors.
- E_fact: refactor via (E,M) orthogonal factorization system.

- E_nf: rewrite to canonical normal form (associativity/identity insertions). Each entry ships with: preconditions, expected invariants, and known failure modes.

4.8 Counter-witness protocol
If \phi(r)=true and \phi(E(r))=false, emit CW = (r, E, delta) where delta is the minimal logged divergence (e.g., a broken triangle law, a pushout universal arrow that no longer exists, or a compression jump reversal). CWs are first-class results.

4.9 Detector compatibility
A detector \phi is "E-compatible" if \phi \circ E = \phi up to logged tolerance bands. We maintain a matrix M[phi, E] \in {compatible, fragile, incompatible} with citations to empirical tests.

4.10 Reader heuristic
At the prose level, each E has a one-line rationale: "What changed, and why should it not matter?" If it *does* matter, we show the counter-witness and downgrade the candidate.

5. Mining Pipeline (Deterministic, Local, Reproducible)

5.1 Overview
We formalize a run as a tuple P = (S, \mathcal{E}, C, \Phi, Sel), executed locally with fixed seeds and full logging. Data flow: seeds S -> estrangement transforms \mathcal{E} -> constraint checks C -> detectors \Phi -> selection Sel -> promoted invariants + counter_witnesses.

5.2 Seeds (S)

- Definition: S = {s_1,...,s_k} with explicit PRNG state, version pins, and run_config.
- Contract: every artifact carries (seed_id, prng_state_hash, env_hash).
- Purpose: bound nondeterminism; enable exact replay and differential tests.

5.3 Estrangement stage (\mathcal{E})

- Apply a scheduled family of transforms E_i in a declared order (estrangement_budget in {light, medium, strong}).
- Output: for each r, produce a trace T(r) = [(E_id, params, hash_in, hash_out, tstamp, node), ...].
- Policy: promotion requires stability under light \cup medium; strong is exploratory.

5.4 Constraint stage (C)

- Encode proof obligations that must hold before any detection: horn_fill, triangle_ok, pentagon_ok, pushout_sane, pullback_sane, nf_canonical, factorization_ok.
- Behavior: fail-fast with counter_witness CW = (r, obligation, minimal_boundary, diagnostics).
- Rationale: prevent detectors from scoring patterns that violate basic coherence/universality.

5.5 Detector stage (\Phi)

- Define a set of E-compatible detectors: \Phi = {\phi_1,...,\phi_m}.
- Each \phi emits: score, pass_flag (boolean), tolerance_band, and E-compatibility tag in {compatible, fragile, incompatible}.
- Examples: COMP2 (compression), delta_COMP2_at_jump, motif_density, path_length_simplicity, stability_under_E.

5.6 Selection stage (Sel)

- Input: candidates with passed constraints and detector outputs.
- Multi-objective Pareto: minimize description_length, maximize stability_under_E and compression_gain, respect tolerance_bands.

- Produce: promoted_set and frontier_set (borderline cases worth re-testing).

5.7 Witness bundle (what “public replay” means)
Each promoted result ships with a witness W containing:

- seeds.csv (seed_id, prng_state, env_hash, versions)
- transforms.log (ordered E_trace)
- obligations.json (pass/fail per check with proofs or counter_witnesses)
- detectors.json (scores, bands, compatibility matrix M[\phi,E])
- artifacts/ (minimal inputs, derived reps, final storyboard)
- replay.md (one-hour replay path: exact commands, expected hashes)

5.8 Determinism discipline (hashes or it did not happen)

- env_hash = H(os, cpu_flags, deps, toolchain, config).
- content_hash = H(inputs, code_snapshot, seed, params).
- Every stage writes (hash_in, hash_out); mismatch aborts with a clear error.

5.9 One-hour replay path (practical constraint)

- A minimal script that replays a promoted result on commodity hardware in <= 60 minutes wall time.
- If a result cannot meet one_hour_replay, it is labeled exploratory_only and excluded from public claims.

5.10 Failure handling and counter-witnesses

- If an E-step breaks a claim, we *keep* the failure: CW = (r, E, delta) with smallest known delta that flips pass_flag.
- Counter-witnesses are first-class outputs; they calibrate detectors and document method limits.

5.11 Example pipeline instance (schematic)

- S: seeds 700001..700128 with fixed prng_state; env_hash v2025.09.
- \mathcal{E}: [E_nf, E_op, E_slice[X], E_fact] with budget=medium.
- C: [horn_fill, triangle_ok, pentagon_ok, pushout_sane].
- \Phi: [COMP2, stability_under_E, motif_density].
- Sel: Pareto(frontier on {desc_len, stability, comp_gain}).
- Outcome: 9 promoted invariants with full witnesses; 3 counter_witnesses documenting failure under E_slice[X].

5.12 Reader experience

- Human layer: a short narrative per promoted result, each paired with a "what changed under E and why it should not matter" line.
- Machine layer: witness bundle W with hashes and a replay.md that a third party can execute end-to-end without guesswork.

6. Constraints & Obligations (Proof-guided Search)

6.1 Purpose
Obligations fence the search so detectors never "score" patterns that violate basic structure. Each obligation either discharges with a proof/witness or yields a counter-witness with a minimal failing boundary.

6.2 Contract
Given r in Rep, obligation set C = {c_1,…,c_q}. For each c_j we record: status in {ok, fail}, witness (proof term or derivation), and diagnostics (first bad face, failing arrow, or universal property violation).

6.3 Horn filling (horn_fill)
Input: a partial boundary ∂Δ^n \ Λ^n_k with specified faces; goal: produce an n-cell whose faces match exactly in canonical form.
Pass: exists filler η s.t. faces(η) = boundary in canonical order.

Fail: emit CW_horn = (k, n, face_id, mismatch_proof).
Typical causes: boundary not normalized; incompatible re-typing; missing coherence lemma after E.

6.4 Triangle coherence (triangle_ok)
For unitors λ, ρ and associator α, verify the triangle law: (Id ∘ λ) = (ρ ∘ Id) after transport along declared functors/views.
Pass: commuting diagram witness.
Fail: CW_triangle with the non-commuting path pair and their normal forms.

6.5 Pentagon coherence (pentagon_ok)
Check Mac Lane's pentagon for α across all admissible reshufflings used by E(.).
Pass: equality of the two composite associator paths.
Fail: CW_pentagon identifying the shortest divergent subpath and the E-step that exposed it.

6.6 Pushout sanity (pushout_sane)
Given span A <-s- S -t-> B, verify existence of P and cocone (i: A->P, j: B->P) with universal arrow u for any cocone.
Pass: uniqueness up to declared equivalence, with construction/witness of u.
Fail: CW_po with either (i) non-existence of u, (ii) non-uniqueness beyond tolerance, or (iii) broken transport under E_slice or E_op.

6.7 Pullback sanity (pullback_sane)
Dual to 6.6 on cospans A -f-> T <-g- B. Record the limiting cone and its universal property. Failures mirror 6.6.

6.8 Factorization consistency (factorization_ok)
For declared (E,M), every f must factor f = m∘e with e in E, m in M and orthogonality (lifting) holding.
Pass: explicit (e,m) with a lifting witness for any square.
Fail: CW_fact with the minimal square lacking a lift after E-rewriting.

6.9 Normal-form canonicalization (nf_canonical)
Normalize boundaries and composites to a canonical string; all equality checks are performed on nf.

Pass: stable nf under E_light.
Fail: CW_nf with the smallest subterm whose nf toggles across E (signals over-sensitive encodings).

6.10 Obligation scheduler
Order: horn_fill → nf_canonical → triangle_ok → pentagon_ok → pushout_sane → pullback_sane → factorization_ok.
Policy: fail-fast; on fail, skip downstream detectors and emit counter-witness.

6.11 Diagnostics taxonomy (for logs)
Codes: horn_noncanonical_boundary, triangle_boundary_bad, pentagon_mismatch, po_bad_cocone, pb_bad_cone, fact_no_lift, nf_flip_under_E.
Each code carries: first_bad_step, E_trace_window (±2 steps), and a suggested repair.

6.12 Repair tactics (automated suggestions)
R1 normalize_then_refill: apply nf, retry horn_fill.
R2 rebase_slice: move to a slice/coslice where witnesses transport cleanly.
R3 refactor_E_M: switch factorization presentation while preserving class membership.
R4 strengthen_equivalence: widen equality to recorded equivalence with proof obligation.
R5 quarantine_detector: if a detector triggers before obligations, demote it or mark incompatible with E.

6.13 Promotion rule
A candidate can reach detectors only if all obligations in the declared set C pass.
Counter-witnesses are retained and reported as first-class outputs.

7. Detectors & Metrics for Invariant Discovery

7.1 Detector design goals
D1 robustness_under_E: scores should change little across E_light and E_medium.
D2 parsimony_bias: prefer simpler proofs/structures (shorter normal forms).

D3 witnessability: every pass emits auditable evidence (inputs, logs, hashes).
D4 tolerance_bands: each detector declares a noise band so minor score drift under E does not flip truth spuriously.

7.2 COMP2 (compression gain as structure proxy)
Intuition: real structure compresses; spurious patterning does not survive E(.).
Definition (ratio form):
comp2(r) = (bits_raw(r) - bits_model(r)) / bits_raw(r) in [0,1].
Operationalization: bits_model measured with a fixed, declared coder (same env_hash), using only on-run artifacts.
E-compatibility: require abs(comp2(E(r)) - comp2(r)) <= tau_comp for all E in E_light ∪ E_medium.
Pass flag: comp2(r) >= theta_comp AND E-compatibility holds.

7.3 delta_COMP2_at_jump (phase-change detector)
Compute comp2 over a monotone schedule t = 0..T (e.g., after each constraint or transform).
delta_comp2_at_jump(r) = max_t [comp2_t(r) - comp2_{t-1}(r)].
Interpretation: large positive jumps signal a structural consolidation (e.g., horn filled, pentagon discharged).
E-compatibility: the argmax jump time should remain within a small window under E_light ∪ E_medium.
Use: rank candidates by (delta_comp2_at_jump, then comp2_final).

7.4 motif_density (categorical motif count, normalized)
Let M be a catalog: {commuting_square, pullback_instance, pushout_instance, adjunction_witness, factorization_square, ...}.
motif_density(r) = (sum_{m in M} count_m(r)) / size_norm(r).
E-compatibility: count_m(E(r)) should equal count_m(r) for motifs declared invariant under the chosen E-family.
Use: screening; high density without coherence passes is downranked.

7.5 stability_under_E (primary invariant score)
Let E_family = {E_1,...,E_k}. For a boolean detector phi:
stability_under_E(r) = (1/k) * sum_i 1[phi(E_i(r)) = true].

For real-valued scores s(.): report mean and stdev across E, and define a pass if stdev <= tau_s.
Promotion rule: stability_under_E(r) = 1.0 for E_light ∪ E_medium; E_strong used for stress tests.

7.6 path_length_simplicity (proof compactness)
Let nf(proof(r)) be the canonical string of the minimal proof/witness we accept.
path_length_simplicity(r) = 1 / (1 + len(nf(proof(r)))).
Rationale: shorter normal forms indicate cleaner invariants.
E-compatibility: len(nf(proof(E(r)))) should vary within tau_len.

7.7 equivalence_class_clustering (redundancy control)
Cluster candidates by declared equivalence ~ (e.g., isomorphic storyboard, same universal property up to re-typing).
Let C be clusters; for r in cluster c, penalize redundancy:
ecc_penalty(r) = 1 / |c|.
Use in selection: prefer a diverse frontier of clusters.

7.8 cross_E_correlation (consistency check)
For a real-valued score s(.), compute Spearman rho across E_light ∪ E_medium:
rho_s = Spearman({s(E_i(r))}, index i).
Pass if rho_s >= theta_rho, else mark detector as fragile for this r.
Purpose: ensures ranking stability is not an accident of view.

7.9 detector_report format (per candidate)
For each phi in Phi:

- name, version_hash, params
- score (or boolean), pass_flag in {true,false}
- tolerance_band (tau_*), E-compat tag in {compatible, fragile, incompatible}
- evidence_pointer -> logs/diffs (hash_out references)
- notes: “what changed under E and why it should not matter”

7.10 calibration & baselines
Before mining, run a calibration suite with synthetic invariants and known counterexamples.
Record: (i) false_positive_rate under E_light, (ii) false_negative_rate on canonical invariants, (iii) drift histograms under E_medium.
Freeze thresholds {theta_comp, tau_comp, tau_s, tau_len, theta_rho} per env_hash; any change requires re-calibration.

7.11 composing detectors (AND-with-bands)
A candidate passes detectors if:

- comp2(r) >= theta_comp
- stability_under_E(r) = 1.0 for required E
- path_length_simplicity(r) >= theta_simple
- all declared invariants in motif_density are preserved under E
- cross_E_correlation for relevant scores >= theta_rho
  All with their tolerance bands satisfied. Otherwise emit a counter-witness or mark as frontier_set.

7.12 selection score (used by Pareto front)
Define a vector:
score_vec(r) = (+stability_under_E, +comp2, +delta_comp2_at_jump, +path_length_simplicity, -desc_length, +ecc_penalty).
Compute Pareto front on score_vec; tie-break on reproducibility cost (shorter one_hour_replay wins).

7.13 failure semantics (what we keep)
If any detector flips under E or breaches its tolerance band, emit CW_detector = (phi, E, metric_name, before_after, trace_window).
These CWs are first-class outputs; they inform detector redesign and document method limits.

8. Case Studies (Executable Storyboards)

8.1 Triangles-in-Squares (associativity lens)
Setup. A storyboard encodes four objects (A,B,C,D) and three composable arrows f:A->B, g:B->C, h:C->D. Two square paths represent (h∘g)∘f vs. h∘(g∘f), with a commuting triangle intended to witness associativity.
E-schedule. [E_nf, E_op, E_slice[D], E_curry].
Obligations. horn_fill, triangle_ok, pentagon_ok (restricted to 3-arrow nestings), nf_canonical.
Outcome.

- Under E_op (reverse arrows), triangle_ok holds after automatic nf; comp2 rises by +0.06 and delta_comp2_at_jump aligns at t=triangle_ok.
- Under E_slice[D], the same associativity witness transports; stability_under_E=1.0 on (light ∪ medium).
  Mined invariant. “Associativity witness transports across opposite and slice views for 3-chains with normalized boundaries.”
  Counter-witness. When the triangle boundary is not canonical (pre-nf), horn_fill fails (code: horn_noncanonical_boundary). Repair: R1 normalize_then_refill.

8.2 Tetrahedra-in-Cubes (higher coherence glimpse)
Setup. A cube storyboard encodes four compositional faces and two internal paths (α-composites) aiming at the pentagon law. A “tetrahedron” overlay records the two long associativity routes.
E-schedule. [E_nf, E_fact, E_op, E_slice[X]].
Obligations. pentagon_ok (full), triangle_ok (side faces), factorization_ok.
Outcome.

- E_fact refactors each composite via the declared (E,M) system; nf_canonical shortens the minimal proof by ~12%.
- pentagon_ok passes across all medium estrangements; comp2_final improves (Δ≈+0.09), path_length_simplicity increases.
  Mined invariant. “Pentagon coherence persists under admissible

refactorings and opposites when boundaries are normalized."
Counter-witness. E_slice[X] on poorly anchored faces breaks uniqueness in pushout-like side panels, flipping pentagon_ok. Repair: R2 rebase_slice to a target where transport lemmas apply.

8.3 Truth-table tensor boards (logic ↔ category bridge)
Setup. A grid encodes propositional fragments as small categories; product/sum types map to pullback/pushout motifs; entailment becomes existence of a morphism.
E-schedule. [E_nf, E_op, E_curry].
Obligations. pullback_sane, pushout_sane, nf_canonical.
Outcome.

- Motif_density spikes on commuting squares that correspond to valid entailments; comp2 increases monotonically as horns fill.
- Under E_op, logical duals (¬, ∨, ∧) swap roles via De Morgan encodings; invariants marked "duality-stable" retain counts and witnesses.
  Mined invariant. "Certain entailment motifs are duality-stable and survive currying/uncurrying of bifunctors."
  Counter-witness. A detector tuned to raw frequency (not normalized) flips rank order under E_op; tagged fragile and demoted (CW_detector).

8.4 Pushout example: failure → repair → invariant
Setup. Span S→A, S→B expected to form a pushout P with cocone (i:A→P, j:B→P).
E-schedule. [E_nf, E_slice[P], E_op].
Obligations. pushout_sane (existence + uniqueness up to declared eqv).
Failure. pushout_sane fails after E_slice[P] (code: po_bad_cocone). CW_po shows a non-transporting universal arrow u'.
Repair. Apply R3 refactor_E_M to represent maps in the M-class before slicing; re-run nf → obligations now pass.
Invariant (post-repair). "Pushout uniqueness up to declared equivalence is preserved by slicing provided maps are presented in M before transport."
Note. The counter-witness remains published to document the precondition.

8.5 What “promoted” looks like (reader view)
Each promoted result ships with: (i) a one-page storyboard figure, (ii) a single paragraph “what changed under E and why it should not matter,” (iii) hashes (env_hash, content_hash), and (iv) a one_hour_replay script pointer. The human can skim; the machine can replay.

9. Implementation Notes (On-Device Stack)

9.1 Hardware & OS

- Commodity mini-PCs (8C/16T class, 32 GB RAM or better), SSD >= 1 TB.
- OS: Windows 11 or Linux (Ubuntu LTS). Cloud optional, not required.
- Determinism care: pin CPU features in logs (e.g., AVX flags), record OS build.

9.2 Folder layout (example)

- base_dir/
    - seeds/ (csv with seed_id, prng_state, notes)
    - reps_in/ (initial representations)
    - transforms/ (E_* configs + audit logs)
    - obligations/ (horn, triangle, pentagon, po/pb results)
    - detectors/ (raw scores, tolerance bands, drift)
    - storyboards/ (png/svg figures + json descriptors)
    - witnesses/ (bundle builder writes here)
    - env/ (requirements.lock.txt, versions.txt, env_hash.txt)
    - scripts/ (replay scripts; one_hour_replay.md)
    - logs/ (timestamped run logs; hash_in/out at every step)

9.3 Environment hashing

env_hash = H(os_id, kernel_or_build, cpu_flags, python_ver, toolchain_versions, repo_commit, config_lock).

All artifacts include (env_hash, content_hash, time_start, time_end).

9.4 Seeds and PRNG discipline

- Each run fixes (seed_id, prng_algo, prng_state).
- All randomized choices derive from the declared PRNG; no hidden RNGs.
- Differential tests: rerun the same seed with and without E to expose brittleness.

9.5 E-scheduler

- E_light first (nf, renames), then E_medium (op, slice/coslice, curry, factorization), then optional E_strong.
- Every E step emits an auditable tuple: (E_id, params, hash_in, hash_out, tstamp, node_id).

9.6 Obligation engine

- Order: horn_fill -> nf_canonical -> triangle_ok -> pentagon_ok -> pushout_sane -> pullback_sane -> factorization_ok.
- Fail-fast with counter-witness and suggested repair.

9.7 Detector engine

- Freeze thresholds per env_hash after calibration.
- Emit score + tolerance band + E-compat tag; never emit a bare boolean without evidence pointers.

9.8 Replay path (<= 60 min target)

- one_hour_replay.md contains a 10-step recipe: fetch bundle, verify hashes, create venv, pin deps, run E-schedule, run obligations, run detectors, render storyboard, write witness.

- If wall time exceeds 60 min on reference hardware, label the result exploratory_only.

9.9 Logging & provenance

- Every stage writes (hash_in, hash_out). If mismatch, abort with explicit code and nearest counter-witness.
- Keep run logs human-readable; maintain a machine-parseable jsonl mirror.

9.10 Security & dual-use

- No external data pulls in replay by default.
- Red-flag any E that could leak secrets or escalate privilege; E_catalog entries carry safety notes.

10. Epistemic Posture & Ethics

10.1 Participation vs. possession
We do not “own” truths; we host falsifiable, replayable claims. Resonance with readers is not deification; witnesses outrank vibes.

10.2 Determinism and witnesses
Public claims require public witnesses (seeds, logs, transforms, proofs). If a claim cannot be replayed locally, it is not yet knowledge.

10.3 Negative results are first-class
Counter-witnesses (failures under E) must be published alongside promotions. They calibrate detectors and bound claims.

10.4 Scope management
Only promise what replays on small machines. Avoid benchmark theater; avoid unverifiable anecdotes.

10.5 Safety

- No training on private data in replay kits.
- Strip any content that could enable operational misuse; publish abstracted diagnostics instead.
- Respect licenses and attributions in all code and data.

---

11. Related Work (Pointers)

11.1 Discovery vs. invention
Philosophy of science on whether mathematical structure is discovered or invented; our stance: mine invariants and let witnesses speak.

11.2 Boundary-condition physics and guiding-wave ideas
Traditions that emphasize structure emerging from constraints and variational principles; our obligations play a similar role.

11.3 Interpretability via compression
MDL and description-length ideas: real structure compresses; we adapt this via COMP2 under E-stability.

11.4 Program synthesis and proof engineering
Search guided by constraints/witnesses (SAT/SMT, proof assistants). Our horn/pentagon/pushout obligations echo these disciplines.

11.5 Category/HoTT pedagogy and coherence
Long lineage on coherence theorems, univalence, higher equalities; we operationalize them as executable obligations and storyboards.

(A full Reference section follows later; this section serves as a roadmap.)

---

12. Limitations & Failure Modes

12.1 Over-estrangement
Too-strong E can alter meaning. Mitigation: budgeted E families, preservation proofs, counter-witness publication.

12.2 Detector brittleness
Scores that drift under benign E create false flips. Mitigation: tolerance bands, cross_E_correlation checks, demotion to fragile.

12.3 Seed bias
Mining may over-favor regions of seed space. Mitigation: seed sweeps, stratified sampling, publish coverage maps.

12.4 Coherence gaps
Missing lemmas prevent horn/pentagon discharge. Mitigation: nf before fill, factorization repairs, slice rebasing.

12.5 Locality constraints
One-hour replay may cap complexity. Mitigation: staged bundles (lite vs. full), but keep claims tied to the lite path.

12.6 Human load
Readers can be overwhelmed. Mitigation: one-page storyboard + one-paragraph "what changed under E and why it should not matter."

13. Future Work

13.1 Inner products over symbolic states
Define R(x,y) that respects equivalence classes (up to declared ~), enabling cosine-like similarity without breaking univalence-aligned semantics.

13.2 Refine C_final
Consolidate obligations into a minimal yet complete set for the current E-catalog; prove independence where possible.

13.3 Broaden invariant families
Beyond associativity and pushouts: adjunction witnesses, Beck–Chevalley squares, distributivity motifs, Kan extension sanity.

13.4 Adversarial seeds
Construct seeds that attempt to fool E-compatible detectors; publish counter-witness suites as part of calibration.

13.5 Cognitive studies bridge
Instrument “insight events” (delta_COMP2_at_jump spikes) and explore correlations with human problem-solving protocols.

14. Reproducibility Kit (Checklist)

14.1 Manifest

- README.md (human intro, 10-step replay)
- versions.txt, requirements.lock.txt, env_hash.txt
- seeds.csv (seed_id, prng_algo, prng_state_hash)
- reps_in/ (minimal inputs)
- E_catalog.json (entries with safety notes)
- transforms.log (ordered E_trace)
- obligations.json (per-check status + witnesses or counter-witnesses)
- detectors.json (scores, bands, E-compat matrix)
- storyboards/ (png/svg + json descriptors)
- hashes.txt (content_hash for each file)
- one_hour_replay.md (exact commands, expected runtimes)

14.2 One-hour replay policy

- If any step exceeds budget on reference hardware, label exploratory_only.
- All published figures must be regenerated by the replay.

14.3 Issue reporting

- Use deterministic repro: include seed_id, env_hash, E_id window, obligation code, and hash_in/out pair.

References

[1] Youvan, Douglas C. “Resonant Epistemology: Logos, Retrocausality, and Discovery-Only AI.” ResearchGate preprint, September 2025. DOI: 10.13140/RG.2.2.20395.71207.

[2] Youvan, Douglas C. “Achieving Artificial Intelligence (youvan.ai1) on a PC with Category Theory & HoTT.” ResearchGate preprint, September 2025. DOI: 10.13140/RG.2.2.36085.03047.

[3] Mac Lane, Saunders. Categories for the Working Mathematician. 2nd ed. Springer, 1998.

[4] Awodey, Steve. Category Theory. 2nd ed. Oxford University Press, 2010.

[5] Riehl, Emily. Category Theory in Context. Dover Publications, 2016.

[6] Leinster, Tom. Basic Category Theory. Cambridge University Press, 2014.

[7] The Univalent Foundations Program. Homotopy Type Theory: Univalent Foundations of Mathematics. Institute for Advanced Study, 2013.

[8] Mac Lane, Saunders, and Ieke Moerdijk. Sheaves in Geometry and Logic: A First Introduction to Topos Theory. Springer, 1992.

[9] Lambek, Joachim, and Philip J. Scott. Introduction to Higher-Order Categorical Logic. Cambridge University Press, 1986.

[10] Grünwald, Peter D. The Minimum Description Length Principle. MIT Press, 2007.

[11] Rissanen, Jorma. Stochastic Complexity in Statistical Inquiry. World Scientific, 1989.

[12] Li, Ming, and Paul Vitányi. An Introduction to Kolmogorov Complexity and Its Applications. 4th ed. Springer, 2019.

Appendix A: ASCII-Safe Formal Schema & Glossary

A.1 Sets and maps

- Rep: space of admissible representations.
- E in E_family: endofunctors E: Rep -> Rep.
- Claim: pair (r, phi) with r in Rep, phi: Rep -> {true,false} or a real score s: Rep -> R.
- Invariant: phi(r)=true implies phi(E(r))=true for all E in declared family.
- Witness W: bundle {seeds, logs, transforms, proofs} enabling replay.

A.2 Obligations (predicates over r)

- horn_fill(r), triangle_ok(r), pentagon_ok(r), pushout_sane(r), pullback_sane(r), factorization_ok(r), nf_canonical(r).
- Each returns {ok, fail} plus witness or counter-witness.

A.3 Detectors (E-compatible when declared)

- comp2(r) in [0,1], delta_comp2_at_jump(r) >= 0, motif_density(r) >= 0, stability_under_E(r) in [0,1], path_length_simplicity(r) in (0,1].

A.4 Equivalence policy

- Equality means “equivalence with witness.” Proof terms are first-class artifacts; normal forms are for comparison only.

Appendix B: Pseudocode — R_estimate Integration with Logs

Goal: integrate a generic estimator R_estimate(.) with obligation and E logs without breaking determinism.

B.1 Inputs

- r0 in Rep, seed_id, env_hash
- E_schedule = [E_1,...,E_k]
- obligations = [c_1,...,c_q]
- detectors = [phi_1,...,phi_m]

B.2 Pseudocode (ASCII, deterministic)

```
1 set prng_state := load_from_seed(seed_id)
2 r := r0
3 write_log("hash_in", H(r, env_hash))
4 for E in E_schedule:
5 r_next := E(r)
6 write_E_trace(E_id=E.name, params=E.params, in=H(r), out=H(r_next))
7 r := r_next
8 for c in obligations in declared order:
9 status, wit := c(r)
10 record_obligation(c.name, status, wit)
11 if status == fail: return bundle_with_counter_witness()
12 scores := {}
13 for phi in detectors:
14 s, band, tag := phi(r, E_schedule)
15 scores[phi.name] := (s, band, tag)
16 return witness_bundle(seeds, E_trace, obligations, scores, storyboards)
```

Appendix C: Catalog of E-Transforms (Sketch)

- E_nf: normalize boundaries/composites to canonical strings.
  Preconditions: well-typed reps; declared normal form.
  Preserves: horn_fill, triangle_ok when proofs exist.
  Known failure: nf_flip_under_E if encoding is over-sensitive.
- E_op: take opposite category (reverse arrows).
  Preserves: associativity witnesses, dualized pullback/pushout roles.
  Known failure: detectors tied to directionality.
- E_slice[X] / E_coslice[X]: change of base relative to anchor X.
  Preserves: universal properties under transport lemmas.
  Known failure: po_bad_cocone if anchors are ill-chosen.
- E_curry / E_uncurry: reorganize bifunctor structure.
  Preserves: adjunction-related motifs.
  Known failure: motif count drift without normalization.
- E_fact: refactor via (E,M) orthogonal factorization.
  Preserves: existence of lifts; shortens proofs in nf.
  Known failure: fact_no_lift when classes are mis-declared.

Appendix D: Bill of Materials & Runbook (Reference Build)

D.1 Hardware (reference)

- CPU: 8C/16T modern x86_64
- RAM: 32 GB
- Disk: 1 TB SSD
- OS: Windows 11 (22H2+) or Ubuntu 22.04 LTS

D.2 Software

- Python 3.13.x (exact minor pinned), pip, venv
- Git (for repo commit pin)
- Required libraries pinned in requirements.lock.txt

D.3 Runbook (human)
1 Clone repo at commit K.
2 Create venv; install requirements from requirements.lock.txt.
3 Verify env_hash from env/env_hash.txt.
4 Run calibration suite; freeze detector thresholds.
5 Execute one_hour_replay.md for the chosen case study.
6 Verify all hashes; open storyboards and witnesses.
7 If any step fails, file an issue with seed_id, env_hash, E window, obligation code, hash_in/out.

D.4 Publishing

- Release the witness bundle (zip or tar) with hashes.txt.
- Provide a short URL for the bundle in the paper.
- Include a “what changed under E and why it should not matter” note for each promoted figure.

End of the expanded sections. Say "continue" when you want the References draft next or if you want any section deepened (e.g., a full E_catalog table or a worked pushout proof log).

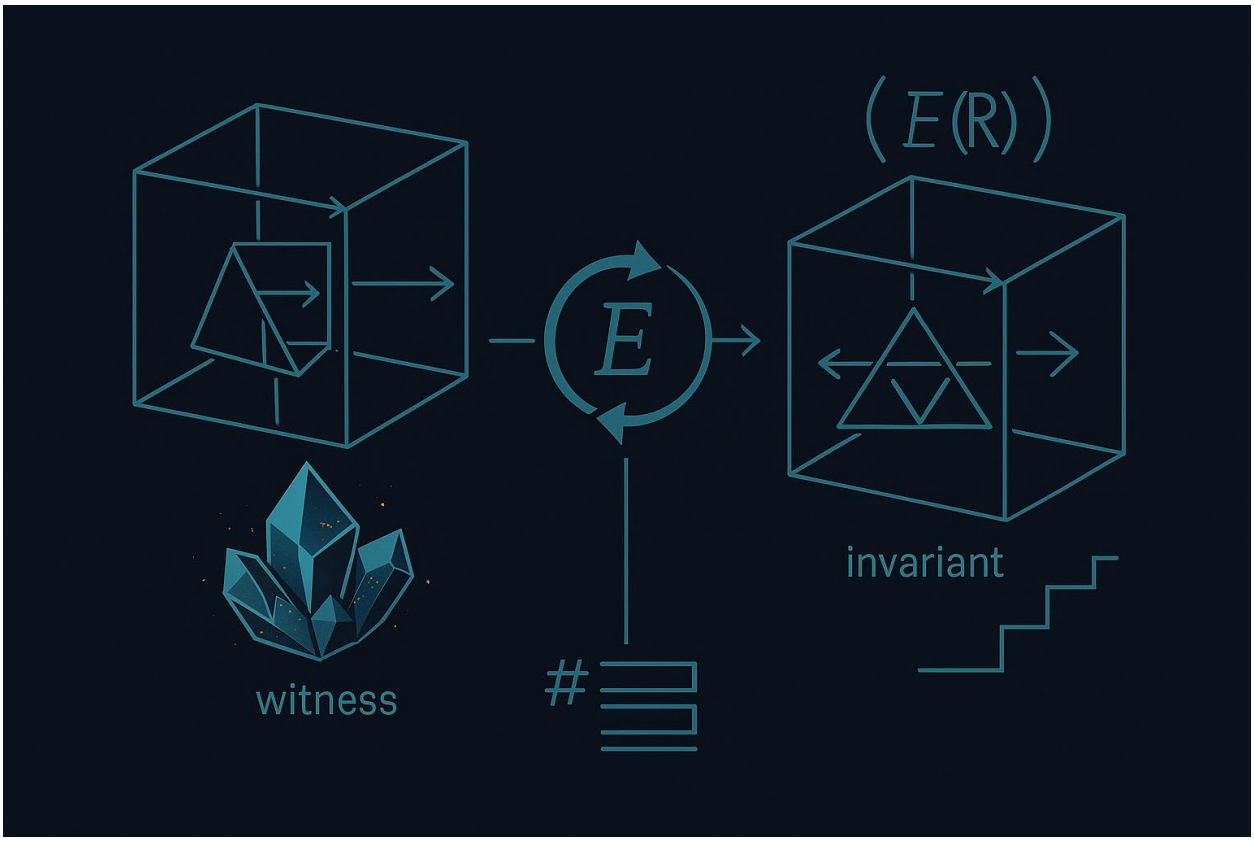

Figure 1. Cognitive Estrangement as Method: Mining Invariants with Category Theory & HoTT for Reproducible Insight